CLAIM AMENDMENTS

Claim Amendment Summary

Claims pending

- Before this Amendment: Claims 1-38.
- After this Amendment: Claims 1-38.

Non-Elected, Canceled, or Withdrawn claims: none

Amended claims: 1, 6-10, 15, 21, 22, 26-28, 30-38

New claims: none

Claims:

1. **(Currently Amended)** A method comprising:

applying a block function to a first data input block from a plurality of data input blocks, wherein the block function comprises a walk on a graph defined by a plurality of matrices; ~~and~~

repeatedly applying the block function to a ~~second~~next data input block from the plurality of data input blocks in accordance with a result of applying the block function to a previous data input block until the block function is applied to a final input block;.

determining a hash value of the plurality of input blocks based on the result provided by the block function applied to the final input block; and

providing the hash value of the plurality of input blocks to a computing environment wherein the hash value facilitates more efficient or more secure data encryption.

2. **(Original)** A method as recited by claim 1, wherein the method is utilized to provide a secure hash function.

3. **(Original)** A method as recited by claim 1, wherein the plurality of data input blocks is formed by dividing an input string.

4. **(Original)** A method as recited by claim 1, wherein each of the plurality of data input blocks has a fixed length.

5. **(Original)** A method as recited by claim 1, wherein one or more of the plurality of data input blocks are padded as needed to provide a fixed length for each of the data input blocks.

6. **(Currently Amended)** A method as recited by claim 1, wherein the graph has a degree *d*~~block function is based on a walk on a graph defined by a plurality of matrices~~.

7. **(Currently Amended)** A method as recited by claim 1, wherein the graph has a degree *d* and the labels are integer labels, wherein each of the integer labels has a value less than or equal to *d*~~further comprising dividing an input string to provide the plurality of data input blocks~~.

8. **(Original)** A method as recited by claim 1, further comprising:

dividing an input string to provide the plurality of data input blocks; and

determining a hash value of the input string, the hash value corresponding to a result provided by the application of the block function to a ~~last~~final data input block.

9. **(Currently Amended)** A method comprising:

providing a graph corresponding to a data input block;

labeling each outgoing edge of every node in the graph with a label; ~~and~~

tracing a path through a plurality of labels on the graph, the path being defined by a sequence of elements within the input block; and

using the tracing of the path for encryption in a computing environment wherein the tracing of the path through the plurality of labels facilitates more efficient or more secure data encryption.

10. **(Currently Amended)** A method as recited by claim 9, wherein the tracing ends at a point that indicates a value of a compression function for a secure hash implementation; and

providing the value of the compression function to the computing environment.

11. **(Original)** A method as recited by claim 9, wherein the graph has a degree *d*.

12. **(Original)** A method as recited by claim 9, wherein the labels are integer labels.

13. **(Original)** A method as recited by claim 12, wherein the graph has a degree *d* and each of the integer labels has a value less than or equal to *d*.

14. **(Original)** A method as recited by claim 9, wherein the input block is a portion of an input string.

15. **(Currently Amended)** In a computing environment, a~~A~~ method comprising:
constructing a table of entries;
setting an initial matrix to an identity matrix;
processing input data as one or more blocks of fixed length;
indexing each block to a generator matrix represented in the table; and
updating the initial matrix.

16. **(Original)** A method as recited in claim 15, wherein the method is utilized to provide a secure hash function.

17. **(Original)** A method as recited in claim 15, wherein advanced encryption standard (AES) is utilized to provide an inter-block function for the blocks.

18. **(Original)** A method as recited in claim 15, wherein the updating is performed by multiplying the initial matrix by the index matrix.

19. **(Original)** A method as recited in claim 15, wherein the table comprises entries for all possible products of a plurality of generator matrices.

20. **(Original)** A method as recited in claim 15, wherein the generator matrix is a free monoid.

21. **(Currently Amended)** One or more computer ~~readable~~storage media ~~storing~~ having computer executable instructions embodied thereon that, when executed in a computing environment, perform the method as recited in claim 15.

22. **(Currently Amended)** A method comprising:

labeling each of a plurality of nodes ~~of a graph with a matrix~~with a matrix, wherein the plurality of nodes make up a graph;

navigating to a next node of the graph; ~~and~~

multiplying ~~the~~each node matrix by at least one of a plurality of generator matrices; and

providing the result of the multiplying each node matrix to a computing environment wherein the result of the multiplying each node matrix facilitates more efficient or more secure data encryption.

23. **(Original)** A method as recited by claim 22, wherein the method is utilized to provide a stream cipher implementation.

24. **(Original)** A method as recited by claim 22, further comprising determining a hash value corresponding to a sequence of intermediate nodes of the graph.

25. **(Original)** A method as recited by claim 22, wherein each of the plurality of generator matrices is a free monoid.

26. **(Currently Amended)** One or more computer ~~readable~~storage media ~~storing~~ having computer executable instructions embodied thereon that, when executed in a computing environment, perform the method as recited in claim 22.

27. **(Currently Amended)** A system comprising:

a processor;

a system memory coupled to the processor;

means for applying a block function to a first data input block from a plurality of data input blocks, wherein the block function comprises a walk on a graph defined by a plurality of matrices; ~~and~~

means for repeatedly applying the block function to a ~~second~~next data input block from the plurality of data input blocks in accordance with a result of applying the block function to a previous data input block until the block function is applied to a final input block;

means for determining a hash value of the plurality of input blocks based on the result provided by the block function applied to the final input block; and

means for providing the hash value of the plurality of input blocks to a computing environment wherein the hash value facilitates more efficient or more secure data encryption.

28. **(Currently Amended)** A system as recited by claim 27, wherein the system is utilized to provide at least one item selected from a group ~~comprising~~consisting of a secure hash function and a stream cipher.

29. **(Original)** A system as recited by claim 27, further comprising means for dividing an input string to provide the plurality of data input blocks.

30. **(Currently Amended)** A system as recited by claim 27, further comprising:

means for dividing an input string to provide the plurality of data input blocks; and

means for determining a hash value of the input string, the hash value corresponding to a result provided by the application of the block function to a ~~last~~final data input block.

31. **(Currently Amended)** One or more computer-~~readable~~ storage media having instructions ~~stored~~embodied thereon that, when executed, direct a machine to perform acts comprising:

applying a block function to a first data input block from a plurality of data input blocks, wherein the block function comprises a walk on a graph defined by a plurality of matrices; ~~and~~

repeatedly applying the block function to a ~~second~~next data input block from the plurality of data input blocks in accordance with a result of applying the block function to a previous data input block until the block function is applied to a final input block;

determining a hash value of the plurality of input blocks based on the result provided by the block function applied to the final input block; and

providing the hash value of the plurality of input blocks to a computing environment wherein the hash value facilitates more efficient or more secure data encryption.

32. **(Currently Amended)** One or more computer~~-readable~~ storage media as recited by claim 31, wherein the method is utilized to provide at least one item selected from a group ~~comprising~~consisting of a secure hash function and a stream cipher.

33. **(Currently Amended)** One or more computer~~-readable~~ storage media as recited by claim 31, wherein the plurality of data input blocks is formed by dividing an input string.

34. **(Currently Amended)** One or more computer~~-readable~~ storage media as recited by claim 31, wherein each of the plurality of blocks has a fixed length.

35. **(Currently Amended)** One or more computer~~-readable~~ storage media as recited by claim 31, wherein one or more of the plurality of data input blocks are padded as needed to provide a fixed length for each of the blocks.

36. **(Currently Amended)** One or more computer~~-readable~~ storage media as recited by claim 31, wherein the graph has a degree *d*~~block function is based on a walk on a graph defined by a plurality of matrices~~.

37. **(Currently Amended)** One or more computer~~-readable~~ storage media as recited by claim 31, wherein the graph has a degree *d* and the labels are integer labels, wherein each of the integer labels has a value less than or equal to *d*~~acts further comprise dividing an input string to provide the plurality of data input blocks~~.

38. **(Currently Amended)** One or more computer~~-readable~~ storage media as recited by claim 31, wherein the acts further comprise:

dividing an input string to provide the plurality of data input blocks; and

determining a hash value of the input string, the hash value corresponding to a result provided by the application of the block function to a ~~last~~final data input block.